Clickbait intensity prediction and its relevance to Fake News

Information Retrieval and Extraction Project

Team 22 **Mentor: Vijaya Saradhi**

Archit Jain (2019101053)
Aryan Jain (2019101056)
Palash Sharma (2019101082)
Pulkit Gupta (2019101078)

Datasets Used

1. Clickbait Intensity

We have used Webis Clickbait Corpus 2017 dataset for predicting clickbait intensity. The dataset includes fields like “postTimestamp”, “postText”, “postMedia”, “targetTitle”, targetKeywords”, “targetDescription”, “targetParagraphs”, “targetCaptions” etc and we have used “postText” as a feature to train the regression model and “truthMean” as the value for clickbait intensity.

2. Fake News Detection

We have used the dataset named fake-real-news-dataset which has two separate files on containing all the articles which are labelled as fake and other one labelled with true, each containing an article title, article text description, its subject along with timestamp of article but we have used title, text and label of article (Fake/Real) along with the clickbait intensity score predicted from model formed in Clickbait Score model.

Prepossessing

1. Cleaning Text

This includes converting text to lowercase, removing non-alphanumeric characters, and using lemmatization and removing ambiguous or incomplete data.

2. Vectorization

Vectorization or word embedding is the process of converting text data to numerical vectors. Later those vectors are used to build various machine learning models. Vectorization is a sophisticated way of including the information contained in the text to machine learning models.

Vectorizers we have used:

a. **spaCy**: A pre-trained word embedding model is loaded, using which the required text is vectorized. The spaCy vectorizer takes a sentence as an input and converts it into vector representation. We used 2 types of pre-trained model. One was small (with just 96 dimensions) and other was large (with 300 dimensions). The model trained on larger dataset produced best results for our case.

b. **Google’s Word2Vec**: Instead of sticking to just one vectorizer, we thought of trying with other vectorizer also namely Google’s Word2Vec embedding model. This model takes one word at a time and returns a particular vector for that word unlike spaCy where we get vector for complete sentence. We used tf-idf score averaging to obtain the vector for the sentence. The vectors obtained from this approach was also of 300 dimensions.

We received better results for spaCy vectorizer.

Algorithms Used

1. Clickbait Intensity

Feature: Vector of “postText” of the article.

We have tried multiple regression models such as:

- Linear Regression (LR)
- Ridge Regression (RR)
- Random Forest Regression (RFR)
- Adaboost Regression (ABR)
- Gradient Boosting Regression (GBR)
- Xtreme Gradient Boosting Regression (XGBR).

Later we also tried Stacking up our model with distinct set of estimators and at last we got our best model with stacking GBR, RR, RFR, ABR and XGBR using LR as our final estimator.

2. Fake News Detection

Features used: vector of title of the article, vector of article text and clickbait score predicated by our Clickbait model.

We have tried multiple classification model such as:

- Logistic Regression (LR)
- Random Forest Classifier (RFC)
- Adaboost Classifier (ABC)
- Xtreme Gradient Boosting Classifier (XGBR)
- Decision Tree (DT)
- Linear SVM (LSVM)
- Naïve Bayes (NB).
- K-Nearest Neighbours Classifier (KNN)

Later we have also tried Stacking up our model with DT, XGBR, RFC, LSVM, ABC as estimators using LR as final estimator.

Evaluation Metrics

- We have used Mean-Squared Error (MSE) as a metrics for evaluating clickbait intensity predicted by regression models.
- We have used F1 score as the main metrics for evaluating fake news predicted by classification models along with accuracy.

Code Link

- GitHub Repo: https://github.com/Architjain128/Clickbait-FakeNews
- Dataset 1: https://zenodo.org/record/5530410#.YXrqehzhU2w
- Dataset 2: https://github.com/laxmimerit/fake-real-news-dataset/tree/main/data

Findings

1. Clickbait

We found that stacking GBR, RR, RFR, ABR and XGBR using LR as our final estimator works best for predicting clickbait intensity.

Algorithm	Number of articles in used dataset sample	
	10k	20k
GBR	0.038497546	0.037490235
RR	0.042002604	0.041711983
LR	0.042178942	0.041664502
RFR	0.157036328	0.159607221
ABR	0.159383018	0.154285207
Stacking (GRB, RR, LR, RFR, ABR)	0.038154828	0.037199447
XGBR	0.037905108	0.036811638
Stacking (GRB, RR, LR, RFR, ABR, XGBR)	**0.036515837**	**0.033900611**

Table 1: MSE values for different algorithms used for different sample sizes

2. Fake News

We observed that LR and PA performed better among all the models in predicting fake news. LR performed better than PA when dataset was without clickbait score and PA performs better than LR in case of with clickbait score.

From the table below we can see that in most of the cases using clickbait score as one of the features gives better f1 score compared to when clickbait score is not used as a feature, but this is not always the case.

Algorithm	Number of articles in used dataset sample					
	5k		10k		20k	
	With Clickbait Score	Without Clickbait Score	With Clickbait Score	Without Clickbait Score	With Clickbait Score	Without Clickbait Score
LR	0.9838	0.9838	0.9868	0.9868	0.9827	0.9820
PA	0.9869	0.9740	0.9868	0.9732	0.9837	0.9664
LSVM	0.9830	0.9890	0.9898	0.9838	0.9816	0.9803
RFC	0.9487	0.9442	0.9641	0.9527	0.9431	0.9430
NB	0.9352	0.9273	0.9292	0.9404	0.9151	0.9124
XGBC	0.9522	0.9582	0.9569	0.9559	0.9542	0.9544
ABR	0.9692	0.9472	0.9796	0.9799	0.9707	0.9686
KNN	0.9502	0.9491	0.9428	0.9418	0.9444	0.944
DT	0.9001	0.8966	0.8925	0.8904	0.8780	0.8794
Stacking	0.9696	0.9696	0.9775	0.9775	0.9793	0.9795

Table 2: F1 scores for different algorithms used for different sample sizes

Algorithm	Number of articles in used dataset sample					
	5k		10k		20k	
	With Clickbait Score	Without Clickbait Score	With Clickbait Score	Without Clickbait Score	With Clickbait Score	Without Clickbait Score
LR	0.984	0.984	0.987	0.987	0.982	0.981
PA	0.987	0.974	0.987	0.973	0.983	0.966
LSVM	0.984	0.990	0.990	0.984	0.982	0.981
RF	0.947	0.944	0.964	0.953	0.942	0.943
NB	0.934	0.926	0.929	0.940	0.913	0.912
XGBC	0.952	0.958	0.956	0.955	0.952	0.953
ABR	0.970	0.950	0.980	0.980	0.971	0.970
KNN	0.952	0.951	0.943	0.942	0.943	0.943
DT	0.901	0.898	0.890	0.889	0.874	0.876
Stacking	0.970	0.970	0.977	0.977	0.979	0.979

Table 2: Accuracy for different algorithms used for different sample sizes

**Green colour indicates improvement of Fake news classification if we use clickbait score*

Changes from scope document

- Initially we had planned to use NELAGT-2019 and NELAGT-2020 datasets for fake news detection but the labelling on that dataset assumed that all the articles from a particular agency we either reliable or unreliable (or mixed). So, we instead went for a dataset that labelled articles independently of being real or fake.

- We have used all the models mentioned in the scope document for regression along with Xtreme Gradient Boosting Regression and stacking up the models.

- For fake news prediction, we have used classification models like LR, DT, XGBR, RFC, LSVM, ABC, NB along with K-Nearest Neighbours that we mentioned in the scope document. And we have tried stacking up the models.

- In the scope document we have mentioned that we will predict clickbait intensity and fake news prediction independently and will find the relevance after combining the results from both. But now we have first predicted clickbait intensity using regression model and then using the best model (with lowest MSE) predicted the clickbait intensity on the fake news dataset and used clickbait intensity as one of the features for predicting fake news to find the relevance between two.